

# Программирование Python

Структуры данных.  
Черновик

Кафедра ИВТ и ПМ

2018



# План

## Прошлые темы

## Пользовательские структуры данных

- Концепция записи

- Запись как кортеж

- Запись как список

- Словарь

- Запись как словарь

- Классы



# Outline

## Прошлые темы

## Пользовательские структуры данных

Концепция записи

Запись как кортеж

Запись как список

Словарь

Запись как словарь

Классы



# Прошлые темы

- ▶ Что такое тип?
- ▶ Какие простые типы есть в Python?
- ▶ Какие составные типы есть в Python?

# Outline

## Прошлые темы

## Пользовательские структуры данных

Концепция записи

Запись как кортеж

Запись как список

Словарь

Запись как словарь

Классы



# Логически связанные данные

Иногда требуется работать одновременно с несколькими отдельными но логически связанными переменными.

Например координаты точки на плоскости представляются двумя переменными.

Это вызывает большие неудобства если таких логически связанных переменных становится много. Например описания погоды (температура, облачность, осадки) в определённом городе, в определённый день.

Неудобства проявляются особенно сильно если нужно создавать ещё одну группу таких переменных для хранения данных, например для другого города.



# Outline

Прошлые темы

**Пользовательские структуры данных**

Концепция записи

Запись как кортеж

Запись как список

Словарь

Запись как словарь

Классы



# Запись

Логически такие наборы данных можно описывать в виде набора полей.

Предположим требуется хранить данные о погодных условиях (температура, облачность, осадки) в определённом городе, в полдень на определённую дату.

Данные о погоде:

Город

ДеньГода

Температура

Облачность

Осадки

Такое представление логически сгруппированных данных будем называть *записью*.





# Концепция записей

							
ВЫПИСЬ ИЗЪ МЕТРИЧЕСКОЙ КНИГѢ, ЧАСТЬ ПЕРВАЯ, О РОДИВШИХСЯ, ЗА ГОДЪ.							
СВѢДѢНІЯ О РОДИВШЕМСЯ		МѢСЯЦЪ И ДЕНЬ	Имя родившагося.	Званіе, имя, отчество и фамилія родителей и каковаго крестнаго дѣдіа.	Званіе, имя, отчество и фамилія крестныхъ родителей.	Кто совершилъ таинство крещенія.	Рисованная или печатная печать по мѣсту.
Мѣсяцъ	День	Мѣсяцъ					
41	25	31	Марія	Анна Ново-Георгиевны пригородной Свѣтлицы? Николаевны Мико- новъ Синайской и законная се- рафа Елизавета Михайловна, обо православныя.	Анна Михайловна Свѣтлицы Павелъ Андреевичъ Удѣльничъ и Анна Павловна Свѣтлицы и Анна Павловна Екатерина Васильевна и Павелъ.	Свѣтлицы Анна Ми- хаиловна Павелъ Николаевъ и диа- конъ Николай Муромовъ.	
			Свѣдѣніе о рождении ребенка (свѣдѣніе о рождении ребенка) изъ метрической книги за 1914 г. № 149.				
1914 г. 25-го			Анна Ново-Георгиевны, пригородной Свѣтлицы? Николаевны Миконовъ Синайской и законная серафа Елизавета Михайловна, обо православныя.				
			Анна Михайловна Свѣтлицы Павелъ Андреевичъ Удѣльничъ и Анна Павловна Свѣтлицы и Анна Павловна Екатерина Васильевна и Павелъ.				

# Концепция записей

4.07.12	Ковалевская И.С.	9000	Всего тисл. записей	Коллекция
07.12	Завалько А.Г.	2500	Фасель сох. ф.	Завалько
07.12	Беленко А.А.	7000	Портрет Тютчева, кл.м.	Беленко
07.12	Козлов И.Б.	1919	одна тисля, часть от	Козлов
07.12	Козлов И.Б.	283,5	записи Косенцев, три в 50.	Козлов
1.08	Козлов И.Б.	4000	звонки за шом	Козлов
1.08.12	Мухоморов	4000	всего тисл.	Мухоморов
4.06.12	Завалько А.Г.	6250	Мухоморов, всего тисл, тисл.	Завалько
4.06.12	Савельев В.П.	1645	Одна тисля, всего тисл.	Савельев
4.06.12	Тютчев А.П.	4399	Звонки, тисля, всего тисл.	Тютчев
1.06.12	Козлов И.Б.	805	Косенцев, всего тисл.	Козлов
4.06.12	Завалько А.Г.	-11-	-11-11-11-11-	Завалько
4.06.12	Савельев В.П.	530	тисля, тисля, тисля	Савельев
4.06.12	Тютчев А.П.	1500	тисля, тисля, тисля	Тютчев
4.06.12	Тютчев А.П.	378	тисля, тисля, тисля	Тютчев
25.07.12	Ленинградская С.А.	3188	Тисля, тисля, тисля	Ленинградская
5.07.12	Ленинградская С.А.	2000	Все тисля, тисля	Ленинградская
5.07.12	Ленинградская С.А.	1195	Тисля, тисля, тисля	Ленинградская



# Запись

**Запись** (record) — тип данных, набор значений различных типов.

Запись состоит из **полей**. Поле как было отмечено может быть представлено отдельным типом. В поле в том числе может быть другой записью.

Например поле *Дата* представляет собой отдельную запись состоящую из трёх полей.

Данные о погоде:

Город

Дата

год

месяц

день

Температура

Облачность

Осадки



# Запись

В языках программирования записи могут быть представлены разными способами.

В Python запись можно представить с помощью одного из составных типов данных: *кортежа, списка или словаря*.

Другой способ представления записи - тип данных **класс (class)**.

Хотя понятия класса включает в себя ещё и методы (операции производимые с данными), его можно использовать только для хранения данных.



# Outline

Прошлые темы

## Пользовательские структуры данных

Концепция записи

**Запись как кортеж**

Запись как список

Словарь

Запись как словарь

Классы



# Записи с помощью кортежей

# Outline

Прошлые темы

## Пользовательские структуры данных

Концепция записи

Запись как кортеж

**Запись как список**

Словарь

Запись как словарь

Классы



# Записи с помощью списков



# Outline

Прошлые темы

## Пользовательские структуры данных

Концепция записи

Запись как кортеж

Запись как список

**Словарь**

Запись как словарь

Классы



# Словари (dict)

# Outline

Прошлые темы

## Пользовательские структуры данных

Концепция записи

Запись как кортеж

Запись как список

Словарь

**Запись как словарь**

Классы



# Словари (dict)

# Outline

## Прошлые темы

## Пользовательские структуры данных

Концепция записи

Запись как кортеж

Запись как список

Словарь

Запись как словарь

**Классы**



## Класс как контейнер данных

Представим запись *Дата* в виде класса (создадим новый тип данных).

```
class Date:
    day = 1
    month = 1
    year = 1
```

В Python нельзя объявить поле, только определить: привести идентификатор и значение.

Когда будет создана переменная типа *Date* её поля уже будут содержать указанные при определении класса значения.

Если нужно отличать переменную с заполненными значениями полей, от аналогичной незаполненными то в качестве начальных значений используют `None`.



## Класс как контейнер данных

Создание переменной описанного типа Date.

Переменная типа класс называется **объектом** или **экземпляром класса**.

```
my_birthday = Date()  
othder_date = Date()  
d1 = Date()
```

```
type( d1 )  # __main__.Date
```

Каждый описанный класс представляет собой отдельный тип данных.

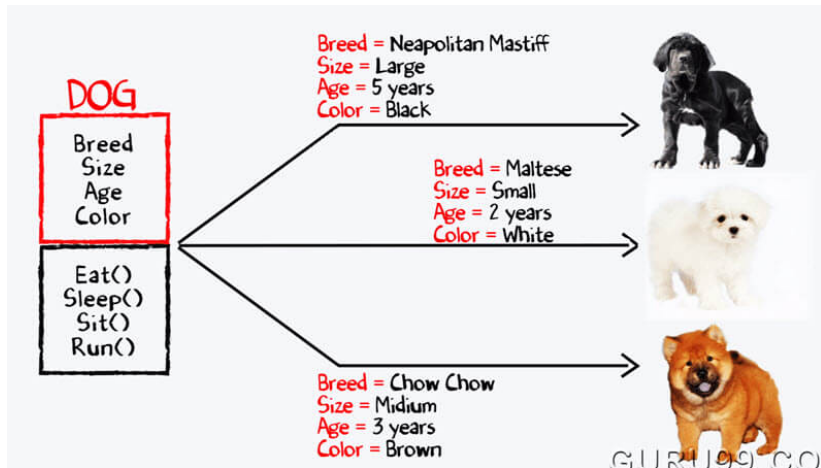
С технической точки зрения имя класса в python включает ещё и имя пространства имён. В приведённом примере пространство имён называется `__main__`<sup>1</sup>

---

<sup>1</sup>Имена переменных начинающиеся и заканчивающиеся символа подчёркивания играют роль служебных ("для внутреннего пользования"), и непосредственное использование таких имён не рекомендуется



# Классы и объекты





# Класс как контейнер данных

Чтобы получить доступ к полям используется **селектор** - оператор "."(точка).

```
my_birthday = Date()
```

```
my_birthday.day = 31
```

```
my_birthday.month = 1
```

```
my_birthday.year = 1956
```



# Класс как контейнер данных

В Python любой тип являются классом, в том числе простые типы.

Однако часто с точки зрения программиста работа с переменными встроенных типов (int, float, tuple и т.д.) выглядит точно так же как и с обычными переменными (в других языках).



# Класс как контейнер данных

```
my_birthday = Date()
```

```
my_birthday.day = 31
```

```
my_birthday.month = 1
```

```
my_birthday.year = 1956
```

*# отдельные экземпляры класса независимы*

```
d = Date()
```

```
d.year = 1984
```

```
print( my_birthday.year )   # 1956
```

```
print( d.year )   # 1984
```



Таким образом для логически связанных наборов данных следует создавать либо классы либо организовывать их с помощью встроенных типов: списков, кортежей, словарей.

Такой подход позволяет логически организовать данные, уменьшить количество отдельных переменных.



# Ссылки и литература

Ссылка на слайды

[github.com/VetrovSV/Programming](https://github.com/VetrovSV/Programming)

